MPT Mixture Models for Continuous Data

Daniel W. Heck

25.02.2020

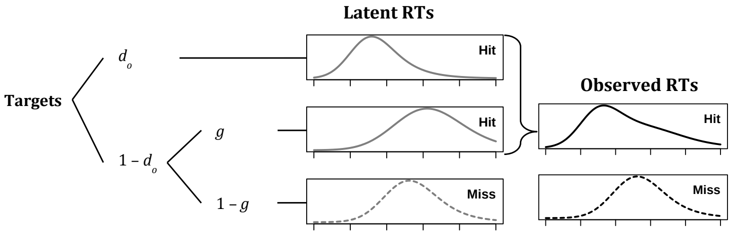
MPT mixture modeling of continuous data

1) MPT-RT: Modeling response times with histograms
   - Heck & Erdfelder (2016)
2) GPT (generalized processing tree): Parametric modeling
   - Heck, Erdfelder, & Kieslich (2018)
3) RT-MPT: Serial-process model for response times
   - Klauer & Kellen (2018)

MPT Models and Continuous Variables

Mixture distribution

- All of the MPT extensions assume mixture distributions for discrete and continuous observations
  - A) Latent RTs: Different processing branches of the MPT model result in different latent distributions $g_j(t)$
  - B) Observed RTs: A mixture distribution, defined as $f(t) = \sum_j p_j g_j(t)$
  - C) The mixture weights p_j are determined by the MPT structure (= branch probabilities)

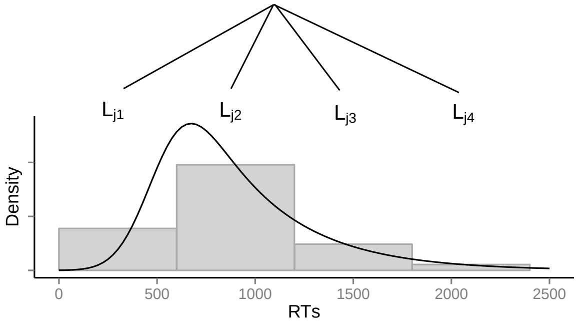


MPT-RT: Modeling response times with histograms

- Heck, D. W., & Erdfelder, E. (2016). Extending multinomial processing tree models to measure the relative speed of cognitive processes. *Psychonomic Bulletin & Review, 23*, 1440–1465.
- Heck, D. W., & Erdfelder, E. (2017). Linking process and measurement models of recognition-based decisions. *Psychological Review, 124,* 442–471.

Histogram-Based Approach (Heck & Erdfelder, 2016)

- Categorize RTs into discrete bins (Yantis, Meyer, & Smith, 1991)
  - Example: “Very fast”, “fast”, “slow”, “very slow”
- State-specific distributions are modeled by the parameters L_{jb}:
  - $L_{jb} =$ height of the histogram bins
  - $L_{jb} =$ probability that state j results in observation in the b-th interval



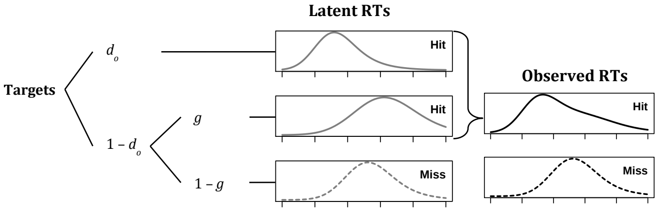
Generalized Processing Tree Models

- Heck, D. W., Erdfelder, E., & Kieslich, P. J. (2018). Generalized processing tree models: Jointly modeling discrete and continuous variables. *Psychometrika.*

Generalized Processing Tree Models (Heck, Erdfelder, & Kieslich, 2018)

Generalized processing tree (GPT) models

- Main difference: Parametric assumptions for component distributions
- The type of distribution depends on continuous variable
  - RTs: log-normal, ex-Gaussian, ...
  - Mouse-tracking measures (see below): Normal distribution
  - Neuro-psychological measures: ...
- The distributions are described by parameters $\boldsymbol{\eta}$
  - Normal distribution: mean and SD
  - ex-Gaussian: mean, SD, and mean of exponential



Generalized Processing Tree (GPT) Models

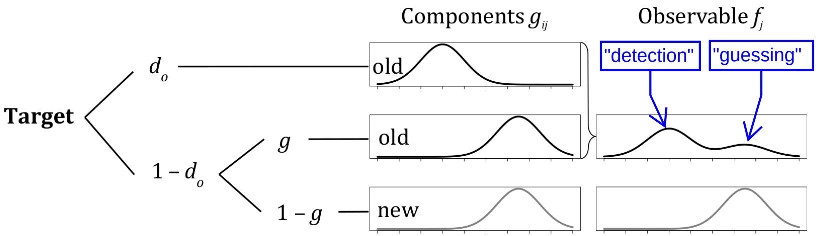
Benefits of the GPT Framework

A) Increased precision in estimating MPT parameters $\boldsymbol{\theta}$
B) Unidentifiable MPT models can become identifiable
C) Flexibility and simplicity

GPTs: Increased Precision

Higher precision of MPT-parameter estimates in GPTs

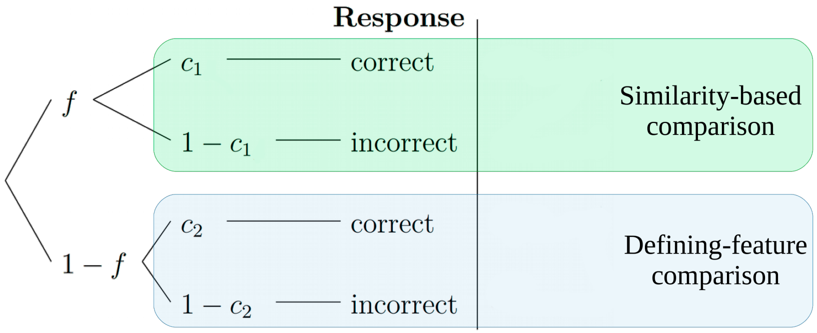
- The more distinct the latent distributions, the smaller the standard error
- Intuition: Continuous variables improve the "classification" which trials belong to which latent processing states
- 2HTM: "Fast RTs are due to detection, slow RTs are due to guessing"



Identifiability of GPT Models

Example: The feature comparison model of semantic categorization

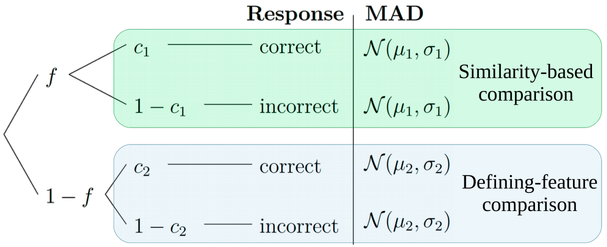
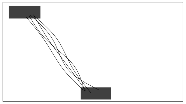
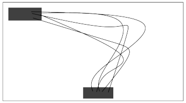
- The theory assumes two different processes
- f = probability of Process 1 (similarity-based comparison)
- c_1 = accuracy of similarity-based comparison
- c_2 = accuracy of defining-feature comparison
- With discrete responses only, the (MPT) model is not identifiable
  - only 1 free category for 3 free parameters



Identifiability of GPT Models

Identifiability of the feature comparison model

- Solution: Assume Gaussian component distributions for continuous variable
  - MAD = maximum absolute deviation in mouse-tracking
- Order constraint for mean parameters: $\mu_c < \mu_d$
  - Interpretation: More direct trajectories/small MADs for similarity-based comparison
- With both discrete *and* continuous data, model is identifiable
  - The different component distribution allow to disentangle the two processes


Response
MAD
f
c_1
correct
$\mathcal{N}(\mu_1, \sigma_1)$
$1 - c_1$
incorrect
$\mathcal{N}(\mu_1, \sigma_1)$
Similarity-based comparison
$1 - f$
c_2
correct
$\mathcal{N}(\mu_2, \sigma_2)$
$1 - c_2$
incorrect
$\mathcal{N}(\mu_2, \sigma_2)$
Defining-feature comparison



Flexibility: GPT Model Specification

GPTs as general-purpose measurement models

- GPTs can be specified as easily as MPTs
  - Implemented in the R package gpt
  - Currently under development: https://github.com/danheck/gpt
  - Define model in a text file similar to EQN
  - Type of latent distribution(s) defined within R (e.g., `latent="normal"`)

GPT version of 2-high-threshold model

```
# Tree ; Categ. ; MPT equation ; mean, SD (normal distr.)
target ; hit    ;  d           ; m_d,  sig
target ; hit    ; (1-d)*g      ; m_g,  sig
target ; miss   ; (1-d)*(1-g)  ; m_g,  sig

lure   ; cr     ;  d           ; m_d,  sig
lure   ; fa     ; (1-d)*g      ; m_g,  sig
lure   ; cr     ; (1-d)*(1-g)  ; m_g,  sig
```

Illustration of the gpt Package

```r
library("gpt")
# data from 2(response bias) x 2(memory strength) design:
# labels: "o30s_cr" = 30% old items / strong memory / correct rejection
head(heck2016, 3)
```

```
##       cat   rt
## 1 o30s_cr 1123
## 2 o30s_cr  671
## 3 o30s_cr  728
```

```r
modelfile <- "models/2htm_exgauss_2x2.txt"
# first lines:
```

```
## # 30% old / strong memory
## lure_s30;     o30s_cr   ;  (1-dn_s)*(1-g30) ; mu,sig,lambda_g_new30
## lure_s30;     o30s_cr   ;  dn_s             ; mu,sig,lambda_dn_s
## lure_s30;     o30s_fa   ;  (1-dn_s)*g30     ; mu,sig,lambda_g_old30
##
## target_s30;   o30s_hit  ;   do_s            ; mu,sig,lambda_do_s
## target_s30;   o30s_hit  ;  (1-do_s)*g30     ; mu,sig,lambda_g_old30
## target_s30;   o30s_miss ;  (1-do_s)*(1-g30) ; mu,sig,lambda_g_new30
##
## # 30% old / weak memory
## lure_w30;     o30w_cr   ;  (1-dn_w)*(1-g30)  ; mu,sig,lambda_g_new30
## lure_w30;     o30w_cr   ;  dn_w              ; mu,sig,lambda_dn_w
## lure_w30;     o30w_fa   ;  (1-dn_w)*g30      ; mu,sig,lambda_g_old30
```

gpt Package: Model Fitting

```
fit <- gpt_fit(x = "cat",         # MPT category
               y = "rt",          # name of continuous variable(s)
               data = heck2016,   # example data for 1 person
               file = modelfile,  # GPT model file
               latent="exgauss",  # family of latent RT distributions
               restrictions=list("dn_s=do_s", "dn_w=do_w"))
fit
```

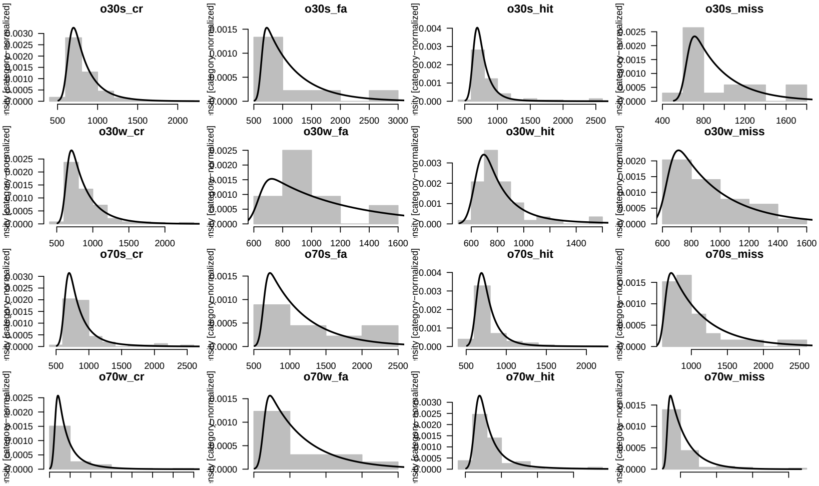
```
##                  Estimate     SE CI.lower CI.upper
## dn_s                0.741  0.027    0.687    0.794
## dn_w                0.477  0.033    0.411    0.542
## g30                 0.189  0.031    0.128    0.250
## g70                 0.261  0.038    0.186    0.335
## lambda_dn_s       172.252 18.159  136.661  207.844
## lambda_dn_w       191.781 34.625  123.916  259.645
## lambda_do_s       128.288 14.735   99.409  157.168
## lambda_do_w       151.728 18.462  115.543  187.912
## lambda_g_new30    311.612 30.782  251.280  371.944
## lambda_g_new70    460.442 36.205  389.481  531.402
## lambda_g_old30    531.220 86.535  361.615  700.824
## lambda_g_old70    517.069 74.575  370.906  663.233
## mu                633.839  5.369  623.316  644.362
## sig                49.194  4.040   41.277   57.112
```

```
test_fit(fit, bins = 4)$test     # Dzhaparidze-Nikulin statistic
```

```
##   statistic df     p.value
## 1  82.50759 42 0.000189941
```

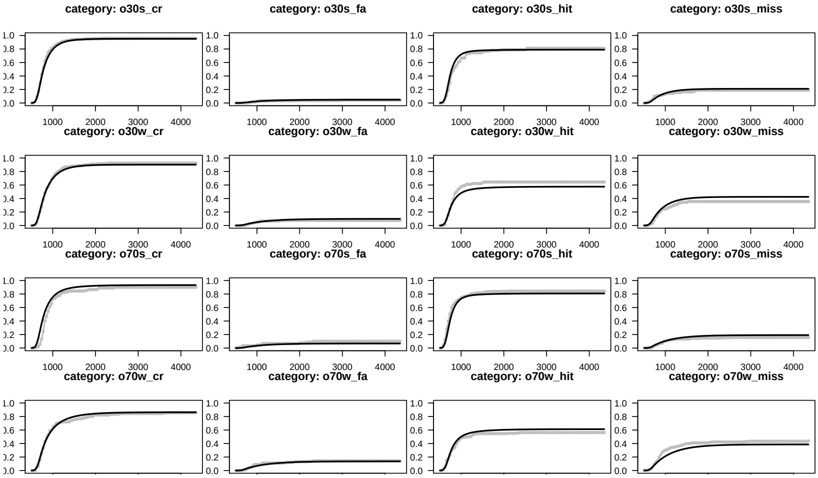
gpt Package: Model Fit

```r
hist(fit)
```



gpt Package: Model Fit

```r
plot(fit)  # cumulative densities
```



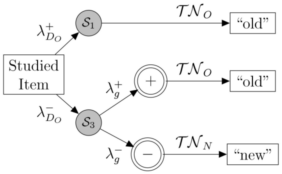
RT-MPT: Serial-process modeling of RTs

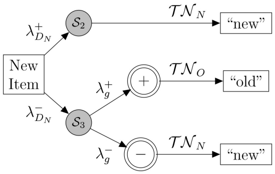
- Klauer, K. C., & Kellen, D. (2018). RT-MPTs: Process models for response-time distributions based on multinomial processing trees with applications to recognition memory. *Journal of Mathematical Psychology, 82*, 111–130.

RT-MPT

RT-MPT Models (Klauer & Kellen, 2018)

- Serial processing assumption: Observed RTs within each MPT branch are the result of a sequence of underlying processes
  1) Time for encoding and response execution
  2) Completion time for each state in the MPT model
  3) Observed RTs in a branch are the sum of encoding and all relevant processing times
- 2HTM components:
  - "Detection RTs": Encoding + Detection
  - "Guessing RTs": Encoding + unsuccessful detection + guessing

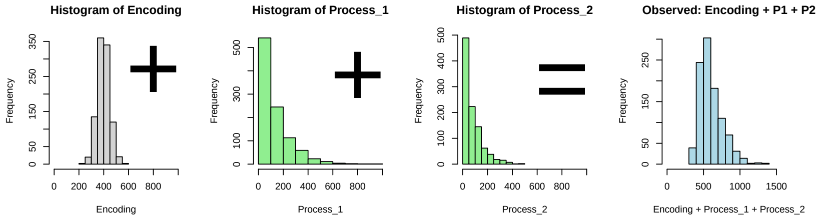




Parametric Assumptions

Illustration of the parametric assumptions

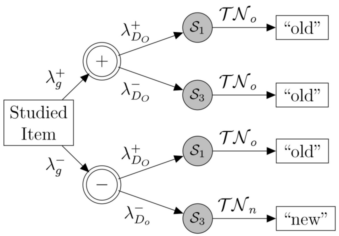
- Encoding and response execution (gray): truncated normal distribution (with mean μ and SD σ)
- Completion times (green): exponential distribution (with rate parameters λ)
- Observed time (blue): sum of encoding and processing times
- Note: Encoding and all completion times are independent

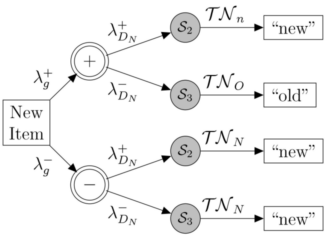


RT-MPT

The ordering of the latent MPT states matters

- The 2HTM permits two possible orders:
  - A) "Detect-Guess Model": First detection, then guessing (previous slides)
  - B) "Default-Interventionist Model": First guessing (by default) and the detection process can intervent (see below)
- With the RT-MPT extension, both versions can be tested against each other

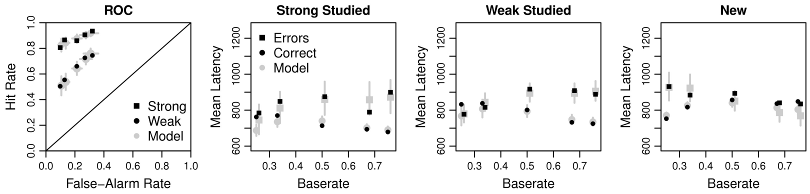




Different 2HTM Versions

Empirical test of different 2HTM versions

- Bayesian hierarchical model (person random effects)
- The Default-Interventionist model fits 5 data sets better
- Effect of manipulations
  - No effect of memory strength on completion time of detection
  - Faster completion times for guessing if response matches the response-bias-condition
- The model fit is satisfactory:



Summary

Summary & Conclusion

RT-Extended MPT Models (Heck & Erdfelder, 2016)

A) No assumptions about shape of latent distributions
B) RT-extended MPT models are also MPT models
C) Testing relative speed of processes (stochastic dominance)

GPT Models (Heck, Erdfelder, & Kieslich; 2018)

A) General approach for modeling discrete and continuous data
B) New tests of psychological theories (e.g., mouse-tracking)
C) MPT parameters estimated more precisely
D) User-friendly software (R package gpt)

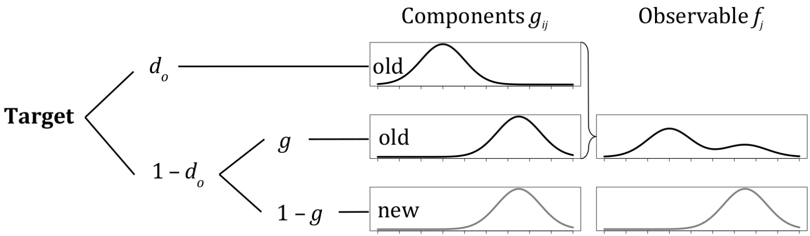
RT-MPT (Klauer & Kellen, 2018)

A) Assumption of serial processing (sum of encoding and completion times)
B) Hierarchical Bayesian

Appendix

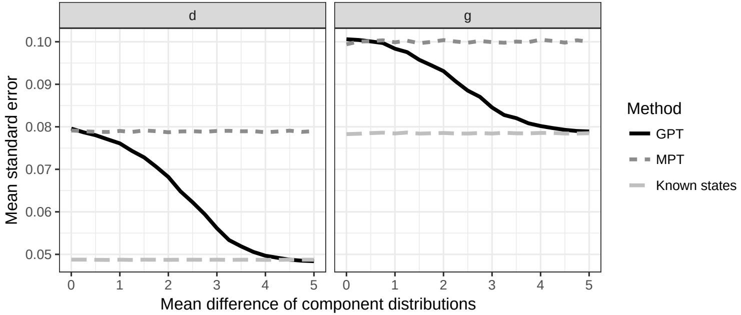
Appendix: GPT Estimates are More Precise

- Higher precision of parameter estimates $\hat{\boldsymbol{\theta}}$
  - The more distinct the latent distributions, the smaller the standard error of $\hat{\boldsymbol{\theta}}$
  - Intuition: Continuous variables improve the classification of trials to latent processing states
  - Upper bound: Precision of MPT model
  - Lower bound: Latent states known



Appendix: GPT Estimates are More Precise

- Simulation of the 2HTM with Gaussian component distributions
  - $\mu^{\text{detect}} = 0$ vs. $\mu^{\text{guess}} = 0, \ldots, 5$
- Results
  - More distinct distributions: smaller standard error of $\hat{\boldsymbol{\theta}}$
  - Upper bound: Precision of MPT model
  - Lower bound: Latent states known



Appendix: Formal Definition of GPT Models

For a vector of discrete responses $\boldsymbol{x}$, a matrix of continuous variables $\boldsymbol{Y}$:

- The **joint distribution** f is a finite mixture
  - For a discrete response x and continuous response(s) $\boldsymbol{y}$:

$$f(x, \boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}) = \sum_{j=1}^{J} \delta_{C_j}(\{x\}) \sum_{i=1}^{I_j} p_{ij}(\boldsymbol{\theta}) g_{ij}(\boldsymbol{y} \mid \boldsymbol{\eta})$$

- **Mixture weights** $p_{ij}(\boldsymbol{\theta})$
  - Identical to MPT-branch probabilities (probability parameters $\boldsymbol{\theta}$)

$$p_{ij}(\boldsymbol{\theta}) = c_{ij} \prod_{s=1}^{S} \theta_s^{a_{ijs}} (1 - \theta_s)^{b_{ijs}}$$

- **Basis distributions** $g_{ij}(\boldsymbol{y} \mid \boldsymbol{\eta})$ for latent states
  - E.g., normal, exGaussian, exWald,... distributions
  - Product-distributions for multivariate continuous data

Appendix: Formal Definition of GPT Models

- GPT models are a set of parameterized distributions:

$$\mathcal{M}^{\mathsf{GPT}}(\Theta = [0,1]^{S_1}, \Lambda \subset \mathbb{R}^{S_2}) = \{f(x, \boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\eta}) \mid \boldsymbol{\theta} \in \Theta, \boldsymbol{\eta} \in \Lambda\}$$

- Likelihood where trials $k = 1, \ldots, K$ fall into disjoint sets M_t that are modeled by $t = 1, \ldots, T$ processing trees

$$L(\boldsymbol{\theta}, \boldsymbol{\eta} \mid \boldsymbol{x}, \boldsymbol{Y}) = \prod_{t=1}^{T} \prod_{k \in M_t} f_t(x_k, \boldsymbol{y}_k \mid \boldsymbol{\theta}, \boldsymbol{\eta})$$

Appendix: GPT Parameter Estimation

Expectation-Maximization (EM) algorithm

- E: Compute expected probabilities $\boldsymbol{z}$ of being in the cognitive states
  - Continuous variables inform the state-vector $\boldsymbol{z}$
- M: Maximize likelihood of continuous parameters given the latent-states $\boldsymbol{z}$

Illustration

- E-step estimates the probability to be in state i in trial k:

$$P(z_k = i \mid \boldsymbol{\theta}, \boldsymbol{\eta}, x_k, \boldsymbol{y}_k) = \frac{P(z_k = i \mid \boldsymbol{\theta}, x_k) g_{ij}(\boldsymbol{y}_k \mid \boldsymbol{\eta})}{\sum_i P(z_k = i \mid \boldsymbol{\theta}, x_k) g_{ij}(\boldsymbol{y}_k \mid \boldsymbol{\eta})}$$

Cat.	RT [ms]	Conf. [1-10]	ERP [mV]	$z_k = 1$	...	$z_k = I$
c_1	551	3	1.324	0.43	...	0.09
c_1	502	1	0.921	0.19	...	0.56
c_2	470	6	2.231	0.30	...	0.00
c_1	733	4	1.010	0.14	...	0.47

Appendix: Identifiability

Identifiability: GPT models with identifiable MPT structure are identifiable if (cf. distribution-free approach):

- Component distributions are observable (Yantis et al., 1991)
- Stepwise deletion of identifiable component distributions (Heck & Erdfelder, 2016)
- Specific matrix has full rank (Heck & Erdfelder, 2016)

Alternative strategy

- Using order constraints to identify GPT models with a nonidentifiable MPT structure
- Label switching of processing paths with component distributions